Soit $\mathcal{G} = \langle X, V, P \rangle$ une grammaire algébrique.

La grammaire $\mathcal{G}$ est *pré-réduite* si :

(1) aucun non-terminal n'engendre le langage vide :

$$\forall S \in V, L_{\mathcal{G}}(S) \neq \emptyset$$

La grammaire $\mathcal{G}$ est *réduite vis-à-vis d'un non-terminal* $T \in V$ si elle est pré-réduite et si :

(2) tous les non-terminaux peuvent apparaître par dérivation à partir de T :

$$\forall S \in V, \exists u_1, u_2 \in (X \cup V)^* \text{ vérifiant } u_1 S u_2 \in \widehat{L}_{\mathcal{G}}(T)$$

Pour réduire $\mathcal{G}$, on peut définir :

(1) $U_0 = X, U_i = U_{i-1} \cup \{S \in V : \exists m \in U_{i-1}^*, S \longrightarrow m \in P\}$ et $U = (\bigcup_{i \geq 0} U_i) \setminus X$ l'ensemble des variables *productives* ;

(2) $W_0 = \{T\}, W_i = W_{i-1} \cup \{S \in V : \exists S' \in W_{i-1}, \exists m_1, m_2 \in (X \cup V)^*, S' \longrightarrow m_1 S m_2 \in P\}$ et $W = \bigcup_{i \geq 0} W_i$ l'ensemble des variables *accessibles* à partir de T.

On obtient une grammaire pré-réduite $\mathcal{G}'$ (*resp.* réduite $\mathcal{G}''$) en supprimant dans $\mathcal{G}$ (*resp.* dans $\mathcal{G}'$) toutes les règles où apparaît une lettre non productive (*resp.* non accessible).

La grammaire $\mathcal{G}$ est *propre* si :

(3) aucune règle n'est de la forme $S \rightarrow \varepsilon$ (ε-règle) ;

(4) aucune règle n'est de la forme $S \rightarrow T$ (règle unitaire).

Pour rendre propre $\mathcal{G}$, on peut définir :

(3) $N_0 = \{S \in V : S \rightarrow \varepsilon \in P\}, N_i = N_{i-1} \cup \{S \in V : \exists m \in U_{i-1}^*, S \longrightarrow m \in P\}$ et $N = \bigcup_{i \geq 0} N_i$ l'ensemble des variables *annulables* ;

(4) la relation $\geq$ par $S \geq T \iff S \xrightarrow{*} T$ et l'équivalence $\sim$ par $S \sim T \iff S \geq T \;\&\; T \geq S$.

La grammaire $\mathcal{G}$ est *sous forme normale de Greibach* (*resp. presque-Greibach*) si on a $P \subseteq V \times XV^*$ (*resp.* $P \subseteq V \times X(X \cup V)^*$).
Pour $V = \{S_1, \ldots, S_r\}$, on pose $V' = \{S'_1, \ldots, S'_r\}$. On construit une suite $\mathcal{G} = \mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{2r+1}$ de grammaires équivalentes :

- $\mathcal{G}_{2i}$ s'obtient en remplaçant dans $\mathcal{G}_{2i-1}$ les règles $S_i \rightarrow S_i m_1 \mid \cdots \mid S_i m_k \mid w_1 \mid \cdots \mid w_p$ par

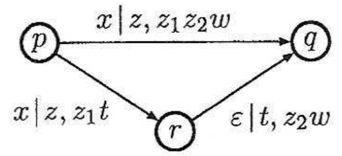
$$\left\{ \begin{array}{lcl} S_i & \rightarrow & w_1 S'_i \mid \cdots \mid w_p S'_i \mid w_1 \mid \cdots \mid w_p \\ S'_i & \rightarrow & m_1 S'_i \mid \cdots \mid m_k S'_i \mid m_1 \mid \cdots \mid m_k \end{array} \right.$$

- $\mathcal{G}_{2i+1}$ s'obtient en substituant dans toute règle de $\mathcal{G}_{2i}$ de la forme $S_j \rightarrow S_i m$ avec $j \neq i$, à cette occurrence de S_i, les membres droits issus de S_i dans G_{2i}.

On peut associer à $\mathcal{G}$ un automate à pile avec un seul état, avec $X \cup V$ pour alphabet de pile, S pour symbole de fond de pile et les transitions $(x/x, \varepsilon)$ pour $x \in X$ et $(\varepsilon/S, \widetilde{m})$ pour $S \longrightarrow m \in P$.
Si $\mathcal{G}$ est sous forme normale de Greibach, il suffit de prendre V pour alphabet de pile, S pour symbole de fond de pile, la transition $(\varepsilon/S, \varepsilon)$ pour $S \longrightarrow \varepsilon \in P$ et les transitions $(a/A, A_k \cdots A_1)$ pour $A \longrightarrow a A_1 \cdots A_k \in P$.

Soit $\mathcal{A} = (Q, \Sigma, Z, T, q_0, z_0)$ un automate à pile.

On peut *standardiser* l'automate $\mathcal{A}$, de sorte que, après avoir dépilé un symbole, chaque transition empile au plus deux symboles.



La *méthode des triplets de Ginsburg* associe à chaque transition une ou plusieurs règles :
- celles qui « dépilent » : $(x, p, z, q, \varepsilon)$ donne $[p, z, q] \to x$;
- celles qui « pilent » : (x, p, z, r, z') donne $[p, z, q] \to x[r, z, q]$;
- celles qui « empilent » : $(x, p, z, r, z_1 z_2)$ donne $[p, z, q] \to x[r, z_2, s][s, z_1, q]$.

Remarquons que si l'automate ne possède pas d'ε-transition, alors la grammaire obtenue est en forme normale de Greibach.

L'automate $\mathcal{A}$ est *déterministe* si :
– pour tout $(p, z, a) \in Q \times Z \times (\Sigma \cup \{\varepsilon\})$, on a $|T(p, z, a)| \leq 1$,
– pour tout $(p, z, a) \in Q \times Z \times \Sigma$, $T(p, z, \varepsilon) \neq \varnothing$ implique $T(p, z, a) = \varnothing$.

On a la chaîne d'inclusions strictes suivante :

{L reconnu par DPDA par pile vide}	$\subsetneq$	{L reconnu par DPDA par états d'acceptation}	$\subsetneq$	{L reconnu par PDA par pile vide et/ou états d'acceptation}
déterministe-préfixe		*déterministe*		*algébrique*

Un langage est déterministe ssi son complémentaire l'est.
Un langage déterministe est non ambigu.